(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 872 986 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
21.10.1998 Bulletin 1998/43

(51) Int Cl.6: H04L 29/02, H04L 29/08, H04L 12/40

(21) Application number: 98302679.0

(22) Date of filing: 06.04.1998

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 15.04.1997 JP 96874/97

(71) Applicant: SONY CORPORATION
Tokyo 141 (JP)

(72) Inventors:
- Kawamura, Harumi, c/o Sony Corporation
  Shinagawa-ku, Tokyo 141 (JP)
- Ludtke, Harald Aaron
  San Jose, CA 95123 (US)

(74) Representative: Pratt, Richard Wilson et al
D. Young & Co,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) Electronic apparatus, information transmitting method thereof, and storing medium

(57) A method for transmitting information between each electronic unit, comprising the steps of (a) transmitting information whose amount does not exceed a predetermined data amount, (b) determining whether or not the predetermined data amount is larger than a desired information amount, (c) when the determined result at step (b) is No, transmitting remaining information for the predetermined amount or less, and (d) repeating the steps (a) to (c) until there is no remaining information.

EP 0 872 986 A2

Printed by Jouve, 75001 PARIS (FR)

Description

The present invention relates to electronic apparatus, information transmitting method thereof, and storing medium. An embodiment of the invention concerns an electronic unit for use with for example an IEEE 1394 serial bus, and to a method of transmitting for example a large amount of data using an asynchronous packet.

A communication system that connects electronic units (hereinafter referred to as units) such as a personal computer, a digital video cassette recorder (hereinafter referred to as DVCR), and a digital television receiver with an IEEE 1394 serial bus and that sends/receives packets of a digital video signal, a digital audio signal, and a control signal therebetween has been proposed.

Fig. 1 shows an example of such a communication system. The communication system comprises a monitor 11, a DVCR 12, and a tuner 13 as units. The monitor 11 and the DVCR 12 are connected with an IEEE 1394 serial bus cable 14. The monitor 11 and the tuber 13 are connected with an IEEE 1394 serial bus cable 15.

In the communication system, an isochronous communication (referred to as ISO communication) for periodically transmitting real time data such as a digital video signal and a digital audio signal between units and an ASYNCHRONOUS communication (ASYNC communication) for non-periodically transmitting commands such as a unit operation control command and a unit connection control command can be performed. For example, a digital video signal and a digital audio signal selected by the tuner 13 can be reproduced as video information and audio information by the monitor 11. Alternatively, such signals can be recorded by the DVCR 12. In addition, a channel selection control command of the tuner 13, an operation mode setup command of the DVCR 12, and so forth can be sent from the monitor 11 to the relevant units through the IEEE 1394 serial bus cables 14 and 15.

In the communication system shown in Fig. 1, there is an AV/C (Audio Visual/Control) command set as commands for controlling AV (Audio Visual) units. In the AV/C command set, a status command for inquiring a status has been defined. In addition, as a response to the status command, status information of a designated unit that is sent back as an operand has been defined.

The data amount of the state may be very large. For example, as shown in Fig. 2, a television broadcast has a hierarchical structure composed of a network layer, a multiplex layer, a service layer, and a component layer. Thus, the data amount of a status command for inquiring each service (broadcast channel) that the digital broadcast tuner is currently selecting may exceed 30 bytes. In the digital broadcast, a plurality of services can be placed on one stream. Thus, a response to an inquiry is required for a plurality of services. Consequently, the data amount of one response may become several hundred bytes.

On the other hand, since the sizes of a command register (buffer) and a response register (buffer) of an FCP (Function Control Protocol) of the IEEE 1394 serial bus are up to 512 bytes, a command packet and a response packet whose sizes exceed 512 bytes cannot be transmitted and received. In addition, it is not assured that a real unit have a buffer that can store data of 512 bytes (the data amounts of currently available buffers are in the range from several ten bytes to one hundred and several ten bytes). When the buffer size is limited, information corresponding to an inquired state cannot be obtained.

An embodiment of the present invention seeks to provide a unit that allows a large amount of data that exceeds the size of a buffer thereof to be obtained and an information transmitting method thereof.

One aspect of the present invention provides a method for transmitting information between electronic units, comprising the steps of (a) transmitting information whose amount does not exceed a predetermined data amount, (b) determining whether or not the predetermined data amount is larger than a desired information amount, (c) when the determined result at step (b) is No, transmitting remaining information for the predetermined amount or less, and (d) repeating the steps (a) to (c) until there is no remaining information.

Another aspect of the present invention provides an electronic unit for communicating with a plurality of units, comprising first means for physically communicating with the plurality of units, buffer means for temporarily storing data that is transmitted by the first means; and controlling means for controlling the first means and the buffer means, wherein the controlling means transmits information whose amount does not exceed a predetermined data amount, determines whether or not the predetermined data amount is larger than a desired information amount, when the determined result is No, transmits remaining information for the predetermined amount or less, and repeats these operations until there is no remaining information.

A further aspect of the present invention provides a storage medium storing a program for an electronic unit for communicating with a plurality of units, comprising first means for physically communicating with the plurality of units, buffer means for temporarily storing data that is transmitted by the first means, and controlling means for controlling the first means and the buffer means, the program causing the controlling means to perform the functions of (a) transmitting information whose amount does not exceed a predetermined data amount, (b) determining whether or not the predetermined data amount is larger than a desired information amount, (c) when the determined result at step (b) is No, transmitting remaining information for the predetermined amount or less, and (d) repeating the steps (a) to (c) until there is no remaining information.

A better understanding of the present invention will become apparent from the following illustrative description thereof which is to be read in connection with the

accompanying drawings; in which:

Fig. 1 is a schematic diagram showing the structure of a communication system using IEEE 1394 serial bus;
Fig. 2 is a schematic diagram showing a hierarchical structure of a television broadcast;
Fig. 3 is a block diagram showing the structure of principal portions of a DVCR according to the present invention;
Fig. 4 is a schematic diagram showing the internal structure of a memory shown in Fig. 3;
Fig. 5 is a schematic diagram showing an example of an object list stored in a descriptor;
Figs. 6A and 6B are schematic diagrams showing an example of information that represents current output signals stored in the descriptor;
Fig. 7 is a schematic diagram showing the structure of **DIRECT SELECT OBJECT** control command corresponding to a tuner sub-unit;
Fig. 8 is a schematic diagram showing the structure of **DIRECT SELECT OBJECT** status command;
Fig. 9 is a schematic diagram showing the structure of a response to the **DIRECT SELECT OBJECT** status command;
Figs. 10A and 10B are flow charts showing a process for checking objects selected in the tuner sub-unit shown in Fig. 3;
Fig. 11 is a schematic diagram showing an example of the structure of a response to the **DIRECT SELECT OBJECT** status command in the case that the capacity of a buffer is sufficient;
Fig. 12 is a schematic diagram showing an example of the content of information of selection_specification;
Fig. 13 is a schematic diagram showing an example of the structure of a response to the DIRECT SELECT OBJECT status command in the case that the capacity of the buffer is insufficient;
Fig. 14 is a schematic diagram showing an example of the structure of a descriptor read command; and
Fig. 15 is a schematic diagram showing an example of the structure of the descriptor read command.

Next, with reference to the accompanying drawings, an illustrative embodiment of the present invention will be described.

Fig. 3 is a block diagram showing the structure of principal portions of an illustrative DVCR according to the present invention. The DVCR comprises a tuner sub- unit 1, a controller 5, a memory 6, and an IEEE 1394 ASYNC block 7.

The tuner sub-unit 1 has an analog broadcast tuner 2 and a digital broadcast tuner 3. The analog broadcast tuner 2 receive a television broadcast signal through an antenna (ANT) 1. The digital broadcast tuner 3 receives a television broadcast signal through an antenna (ANT) 2. A signal of a channel selected by the analog broadcast tuner 2 is sent to a recording portion (DVCR sub-unit) through a sub-unit output plug P1. A stream from a transponder selected by the digital broadcast tuner 3 is sent to a demultiplexer 4. The demultiplexer 4 selects at least one service and sends the selected service to a recording portion and an IEEE 1394 ISO block through a sub-unit output plug P0. The demultiplexer 4 branches service information of the stream to the controller 5. The sub-unit output plugs P0 and P1 are output terminals in the logical meaning and it is not required that they are physical output plugs.

The controller 5 controls the entire DVCR. In addition, the controller 5 creates an object list corresponding to service information received from the demultiplexer 4 and writes the object list to the memory 6. Moreover, the controller 5 sends/receives a command and response to/from another unit through the IEEE 1394 ASYNC block 7 and an IEEE 1394 serial bus 8. Furthermore, the controller 5 writes information of signals that are currently being output from the sub-unit output plugs P0 and P1 to the memory 6.

The memory 6 has a particular area referred to as a descriptor as shown in Fig. 4. In the descriptor, the above-mentioned object list and information of signals that are currently being output are written. Fig. 5 shows an example of the object list. The object list is created corresponding to the multiplex layer, the service layer, and the component layer shown in Fig. 1. Fig. 6A shows the structure of a list (plug list) that shows plugs of the tuner sub-unit and objects that are currently being output from these plugs. This list is referred to as plug tuner object list. Fig. 6B shows a real example of the plug tuner object list. As shown in Fig. 6B, there are two types of object entry describing method. The first method is a detailed type for describing specifications in detail. The second method is a reference type for referencing another list.

The IEEE 1394 ASYNC block 7 assembles a command and a response created by the IEEE 1394 ASYNC block 7 as an ASYNC packet and sends the ASYNC packet to the IEEE 1394 serial bus 8. In addition, the IEEE 1394 ASYNC block 7 disassembles an ASYNC packet received from the IEEE 1394 serial bus 8 into a command and a response and sends the command and the response to the controller 5. At this point, the command and the response are temporarily stored in the buffer memory (that has a transmission buffer and a reception buffer).

Next, a process for checking objects selected by the tuner sub-unit 1 shown in Fig. 3 will be described. First of all, the structure of a command and a response used in the process will be described.

Direct Select Object command as a tuner sub-unit command selects at least one service, multiplexed stream, or component that is being broadcast and outputs the selected service, multiplexed stream, or component to a designated sub-unit plug. A control command designates the selection. A status command in-

quires what is currently being selected.

Fig. 7 shows the structure of a control command. In Fig. 7, **source_plug** represents an output plug of the tuner sub-unit. **subfunction** removes, appends, or replaces a designated object of a designated plug.

tuner_object_selection_specification is a parameter necessary for selection. It is supposed that the amount of information of **tuner_object_selection_specification** is around 10 to 50 bytes. When a command transmitter unit designates a plurality of objects, even if the size of the buffer memory of the IEEE 1394 ASYNC block is not sufficient, the objects can be selected by dividing them into a plurality of responses with **subfunction:append.**

Fig. 8 shows **Direct Select Object** status command. The **Direct Select Object** status command inquires what is currently being output to a designate plug. Fig. 9 shows a response of the **Direct Select Object** status command.

Next, with reference to a flow chart shown in Figs. 10A and 10B, a process for checking objects selected by the tuner sub-unit 1 shown in Fig. 3 will be described.

At step S1, the **Direct Select Object** status command is transmitted. In other words, another unit (for example, the monitor unit) connected to the IEEE 1394 serial bus 8 shown in Fig. 3 places a command as shown in Fig. 8 in an ASYNC packet and sends the resultant ASYNC packet to the IEEE 1394 serial bus 8 through the IEEE 1394 ASYNC block. The packet is input to the IEEE 1394 ASYNC block 7 shown in Fig. 3. The packet is temporarily stored in the buffer memory 9 and then read by the controller 5.

The controller 5 analyzes the received command and checks signals that are currently being output to a designated plug (in this case, the sub-unit plug P0). In other words, the controller 5 checks information of signals that are currently being output with the descriptor stored in the memory 6. As exemplified in Figs. 6A and 6B, the information describes the number of entries of objects for each plug. Thus, the controller 5 reads information of the plug P0, creates a response with the structure shown in Fig. 9, and sends the response back to the relevant unit.

However, the content of the response depends on the size of the buffer memory 9 of the IEEE 1394 ASYNC block 7 and the full length of **tuner_object_selection_specification** in the response. When the size of one **tuner_object_selection_specification** is 30 bytes and four objects are currently being output to the plug P0, the total amount of data of the response becomes 120 bytes.

In this case, when the size of the transmission buffer of the buffer memory 9 is sufficient, the controller sends a response as shown in Fig. 11 back to the relevant unit. In this case, **operand** [0] is stable. With **operand [3]** to **[x]**, information of four **selection_specification** [0] to [4] is sent back to the relevant unit. Fig. 12 shows an example of the content of each **selection_specification.**

On the other hand, when the size of the transmission buffer of the buffer memory 9 is for example 100 bytes, information of four objects cannot be sent back to the relevant unit. Thus, the controller sends a response as shown in Fig. 13 back to the relevant unit. In this case, **operand [1]** is incomplete. The value of **number_of_object_selection_specification of operand [2]** is the number of objects (= 3) that can be sent back rather than the number of objects that are currently being output from the plug P0. With operand [3], information of three **selection_specification [0]** to **[2]** is sent back to the relevant unit.

An ASYNC packet containing the response is received by the command transmitter unit through the IEEE 1394 serial bus 8 (at step S2). The response is sent to the controller through the IEEE 1394 ASYNC block of the command transmitter unit. The controller references the status field of the response (at step S3).

When the status field is stable, as shown in Fig. 11, the response contains information of all objects. Thus, the command transmitter unit completes the process.

On the other hand, when the status field is incomplete, the response contains information of objects that can be sent back as shown in Fig. 13. Thus, information of signals that are currently being output is read from the descriptor stored in the memory 6. In the following description, a process for a packet transmitted between a command transmitter unit and a command receiver unit (the DVCR shown in Fig. 3) is omitted.

The command transmitter unit sends a command for reading a plug list of the descriptor (at step S4). The controller 5 of the command receiver unit reads the plug list as shown in Figs. 6A and 6B from the descriptor stored in the memory 6 and sends the plug list as a response to the command transmitter unit. The command transmitter unit checks **list_id = xx** of plug 0 from the plug list in the response (at step S5). In this case, it is assumed that xx = 0101.

Next, the command transmitter unit transmits a command for checking the number of entries of objects in the plug list of **list_id = xx** (in this case, xx = 0101) to the command receiver unit. The command transmitter unit determines the number n of entries of objects corresponding to the response (at step S6). In this case, it is assumed that n = 4.

At step S7, the command transmitter unit initially sets k = 0 and sends a command for reading an object of a k-th entry of a plug **list_id = xx** of the descriptor to the command receiver unit. Thereafter, the command transmitter unit collects information of the object of the k-th entry corresponding to the response. After the command transmitter unit has collected the information for n entries, it completes the process (from steps S8 to S10).

Next, a command and a responses at step S9 will be described.

Fig. 14 shows an example of the structure of a command **(READ DESCRIPTOR, list_id = xx, entry = k)**

for reading an object of a k-th entry of a flag list id = xx of the descriptor. **data_length = 0** of **operand [5]** represents that the command transmitter unit requires to read all objects with an entry number k.

Fig. 15 shows an example of the structure of a response to the command shown in Fig. 14. **data_length = yy** of **operand [5]** represents the length of data sent with the response. **entry_length** of **operand** [8] represents the length of the object with the entry number k. When the size of the transmission buffer in the buffer memory 9 of the IEEE 1394 ASYNC block 7 shown in Fig. 3 is 100 bytes and the entry length is 30 bytes, information of one entry can be sent with one response. Thus, with **yy = 30** and **zz = 30,** 30 bytes are read from offset address 0000 designated by **operands [6]** and **[7]** and sent. If the size of the transmission buffer of the buffer memory 9 is smaller than 30 bytes, (for example, the size is 10 bytes), with **yy = 10** and **zz = 30**, the offset address is shifted by 10 bytes and sent as three responses.

Thus, with **READ DESCRIPTOR** command, when the command receiver unit cannot send to the command transmitter unit a response with information required by the command transmitter unit, the command receiver unit sends information of the maximum bytes that the command receiver unit can handle back to the command transmitter unit. In addition, since the command transmitter unit can freely designate an address and data length, it can send large amount of data with a plurality of responses.

As described above, according to an embodiment of the present invention, from a unit with a limited size of a transmission buffer, a large amount of data that exceeds the buffer size can be extracted.

Having described a specific preferred embodiment of the present invention with reference to the accompanying drawings, it is to be understood that the invention is not limited to that precise embodiment, and that various changes and modifications may be effected therein by one skilled in the art without departing from the scope or the spirit of the invention.

Claims

1. A method for transmitting information between each electronic unit, comprising the steps of:

   (a) transmitting information whose amount does not exceed a predetermined data amount;
   (b) determining whether or not the predetermined data amount is larger than a desired information amount;
   (c) when the determined result at step (b) is No, transmitting remaining information for the predetermined amount or less; and
   (d) repeating the steps (a) to (c) until there is no remaining information.

2. The method as set forth in claim 1,
   wherein the step (a) is performed corresponding to IEEE 1394 protocol.

3. An electronic unit for communicating with a plurality of units, comprising:

   first means for physically communicating with the plurality of units;
   buffer means for temporarily storing data that is transmitted by said first means; and
   controlling means for controlling said first means and said buffer means,
   wherein said controlling means transmits information whose amount does not exceed a predetermined data amount, determines whether or not the predetermined data amount is larger than a desired information amount, when the determined result is No, transmits remaining information for the predetermined amount or less, and repeats these operations until there is no remaining information.

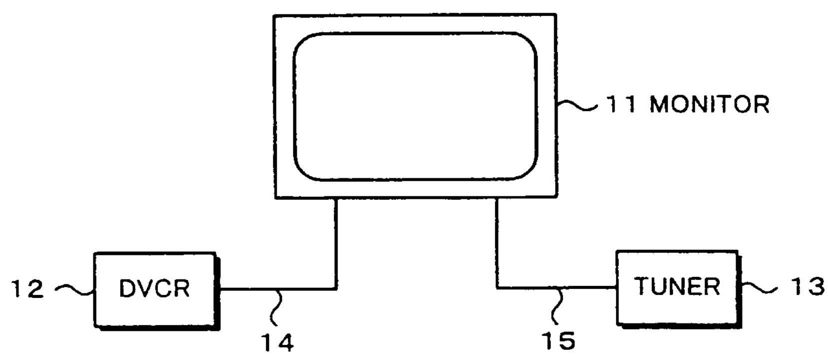
4. The electronic unit as set forth in claim 3,
   wherein said first means communicates with the plurality of units corresponding to IEEE 1394 protocol.

5. A storing medium storing a program for an electronic unit for communicating with a plurality of units, comprising first means for physically communicating with the plurality of units, buffer means for temporarily storing data that is transmitted by the first means, and controlling means for controlling the first means and the buffer means, the program causing the controlling means to perform the functions of:

   (a) transmitting information whose amount does not exceed a predetermined data amount;
   (b) determining whether or not the predetermined data amount is larger than a desired information amount;
   (c) when the determined result at step (b) is No, transmitting remaining information for the predetermined amount or less; and
   (d) repeating the steps (a) to (c) until there is no remaining information.

6. The storing medium as set forth in claim 5,
   wherein the first means communicates with the plurality of units corresponding to IEEE 1394 protocol.

Fig. 1



14,15 : IEEE 1394 SERIAL BUS CABLES

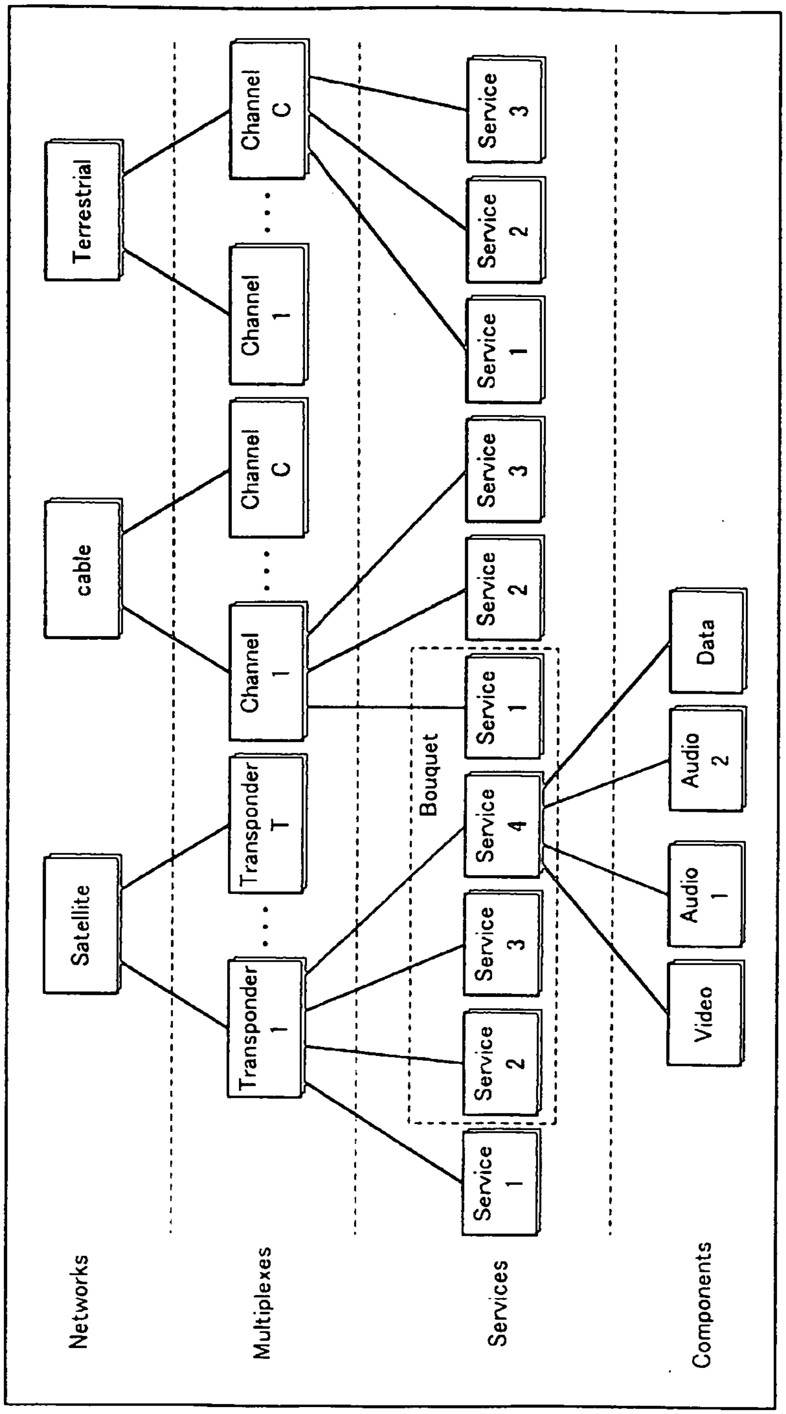

Networks
Multiplexes
Services
Components
Satellite
cable
Terrestrial
Transponder 1
Transponder T
Channel 1
Channel C
Channel 1
Channel C
Service 1
Service 2
Service 3
Service 4
Bouquet
Service 1
Service 2
Service 3
Service 1
Service 2
Service 3
Video
Audio 1
Audio 2
Data

Fig. 2

Fig. 3

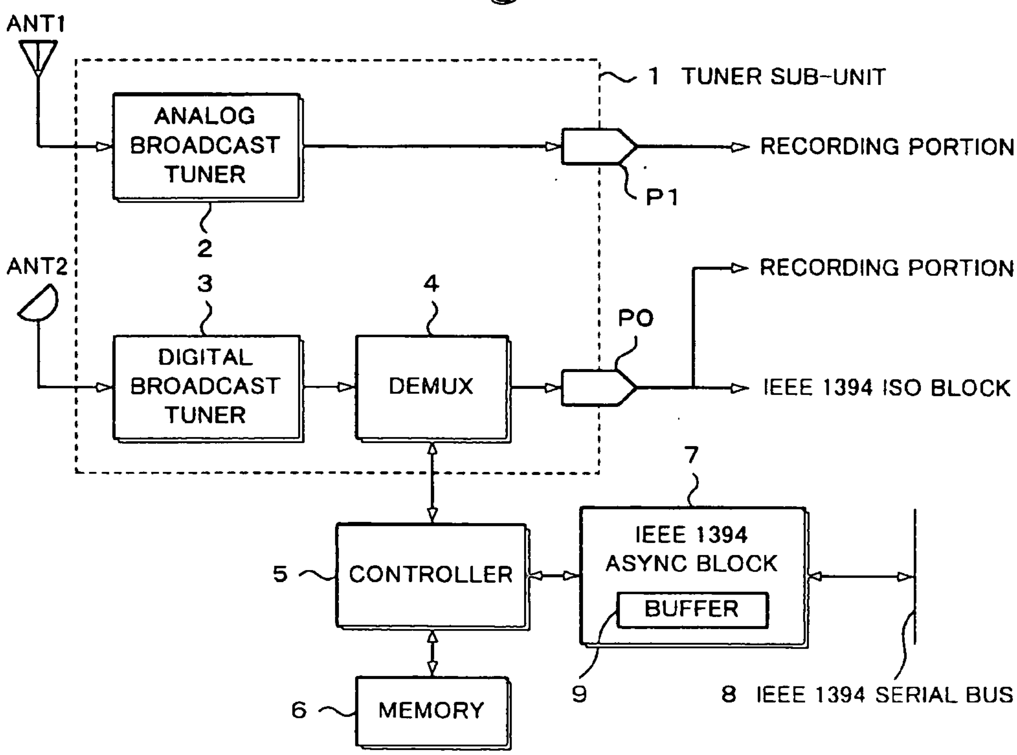

ANT1
1 TUNER SUB-UNIT
ANALOG BROADCAST TUNER
RECORDING PORTION
P1
2
ANT2
3
4
RECORDING PORTION
PO
DIGITAL BROADCAST TUNER
DEMUX
IEEE 1394 ISO BLOCK
7
IEEE 1394 ASYNC BLOCK
5
CONTROLLER
BUFFER
9
8 IEEE 1394 SERIAL BUS
6
MEMORY

Fig. 4

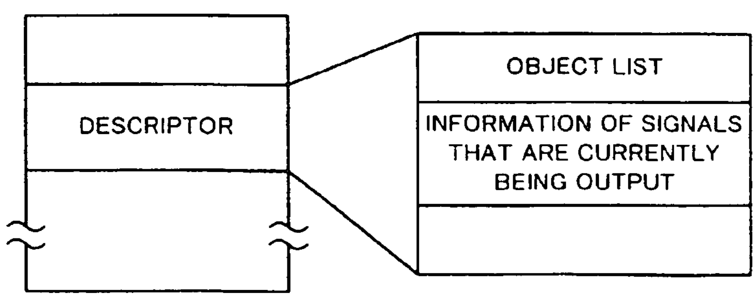

DESCRIPTOR
OBJECT LIST
INFORMATION OF SIGNALS THAT ARE CURRENTLY BEING OUTPUT

Fig. 5

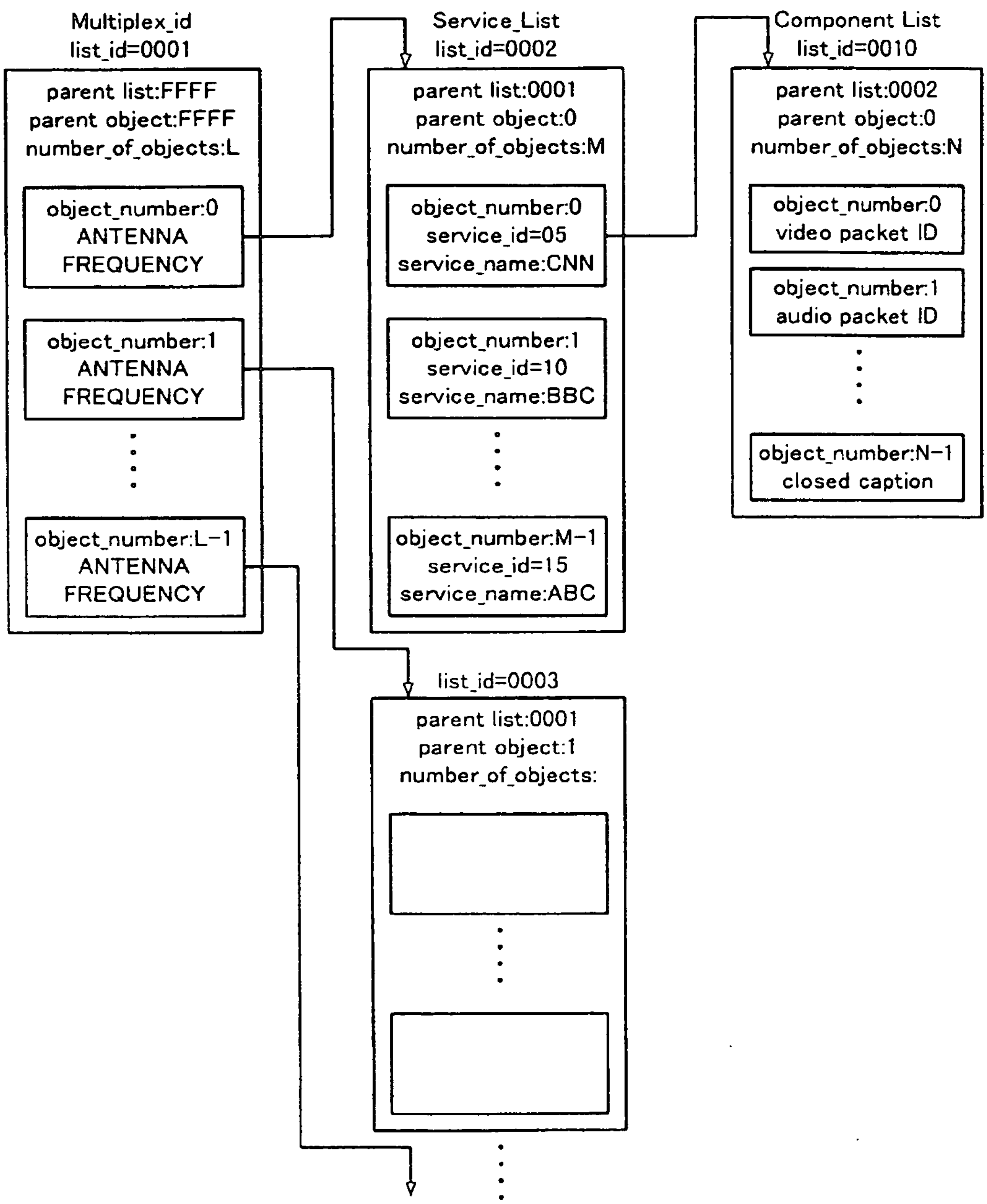

Multiplex_id
list_id=0001
parent list:FFFF
parent object:FFFF
number_of_objects:L
object_number:0
ANTENNA
FREQUENCY
object_number:1
ANTENNA
FREQUENCY
object_number:L-1
ANTENNA
FREQUENCY
Service_List
list_id=0002
parent list:0001
parent object:0
number_of_objects:M
object_number:0
service_id=05
service_name:CNN
object_number:1
service_id=10
service_name:BBC
object_number:M-1
service_id=15
service_name:ABC
Component List
list_id=0010
parent list:0002
parent object:0
number_of_objects:N
object_number:0
video packet ID
object_number:1
audio packet ID
object_number:N-1
closed caption
list_id=0003
parent list:0001
parent object:1
number_of_objects:

Fig. 6A

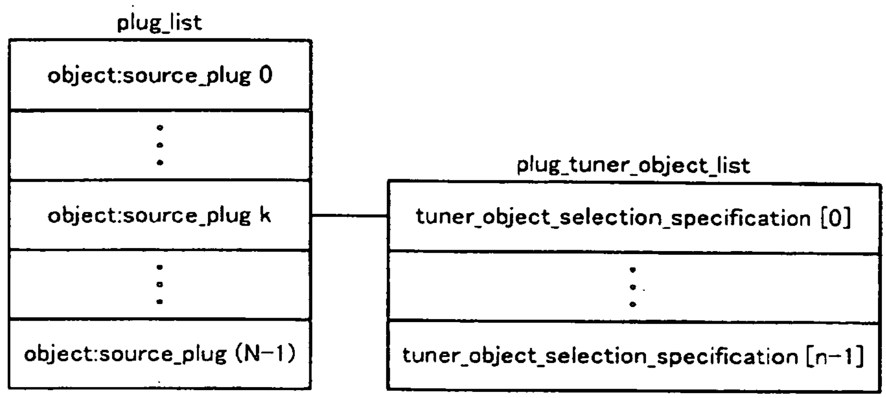

plug_list
object:source_plug 0
object:source_plug k
object:source_plug (N−1)
plug_tuner_object_list
tuner_object_selection_specification [0]
tuner_object_selection_specification [n−1]

Fig. 6B

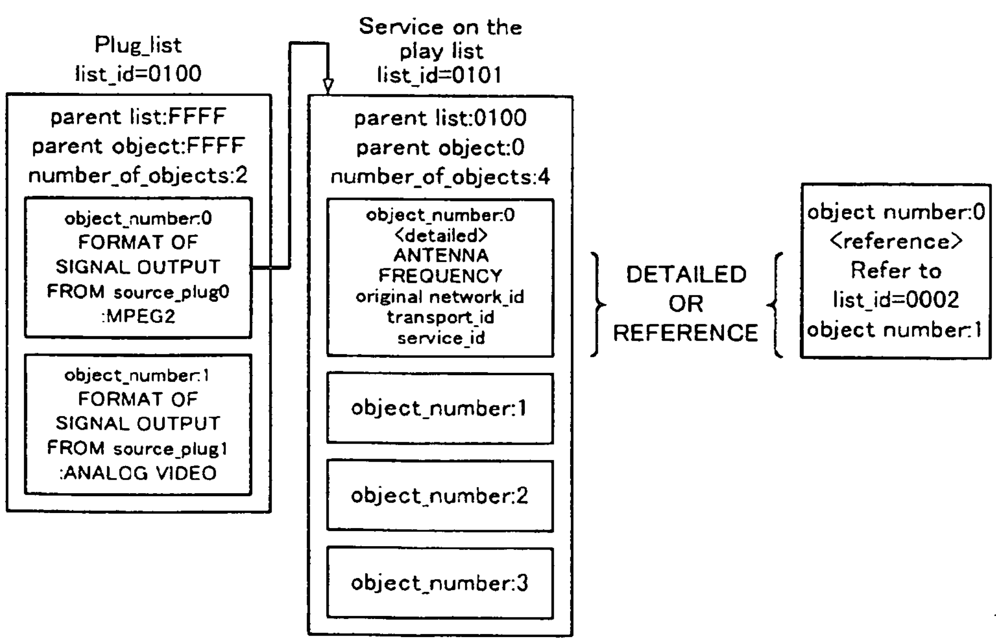

Plug_list
list_id=0100
parent list:FFFF
parent object:FFFF
number_of_objects:2
object_number:0
FORMAT OF
SIGNAL OUTPUT
FROM source_plug0
:MPEG2
object_number:1
FORMAT OF
SIGNAL OUTPUT
FROM source_plug1
:ANALOG VIDEO
Service on the
play list
list_id=0101
parent list:0100
parent object:0
number_of_objects:4
object_number:0
<detailed>
ANTENNA
FREQUENCY
original network_id
transport_id
service_id
object_number:1
object_number:2
object_number:3
DETAILED
OR
REFERENCE
object number:0
<reference>
Refer to
list_id=0002
object number:1

Fig. 7

opcode	DIRECT SELECT OBJECT
operand [0]	source_plug
operand [1]	subfunction
operand [2]	number_of_object_selection spesification (n)
operand [3] ⋮	tuner_object_selection_specification [0]
⋮	⋮
⋮	tuner_object_selection_specification [n−1]

Fig. 8

opcode	DIRECT SELECT OBJECT
operand [0]	(FFh)
operand [1]	subfunction
operand [2]	(FFh)

Fig. 9

opcode	DIRECT SELECT OBJECT
operand [0]	source_plug
operand [1]	status
operand [2]	number_of_object_selection spesification (n)
operand [3] ⋮	tuner object_selection_specification [0]
⋮	⋮
⋮	tuner_object_selection_specification [n-1]

Fig. 10A

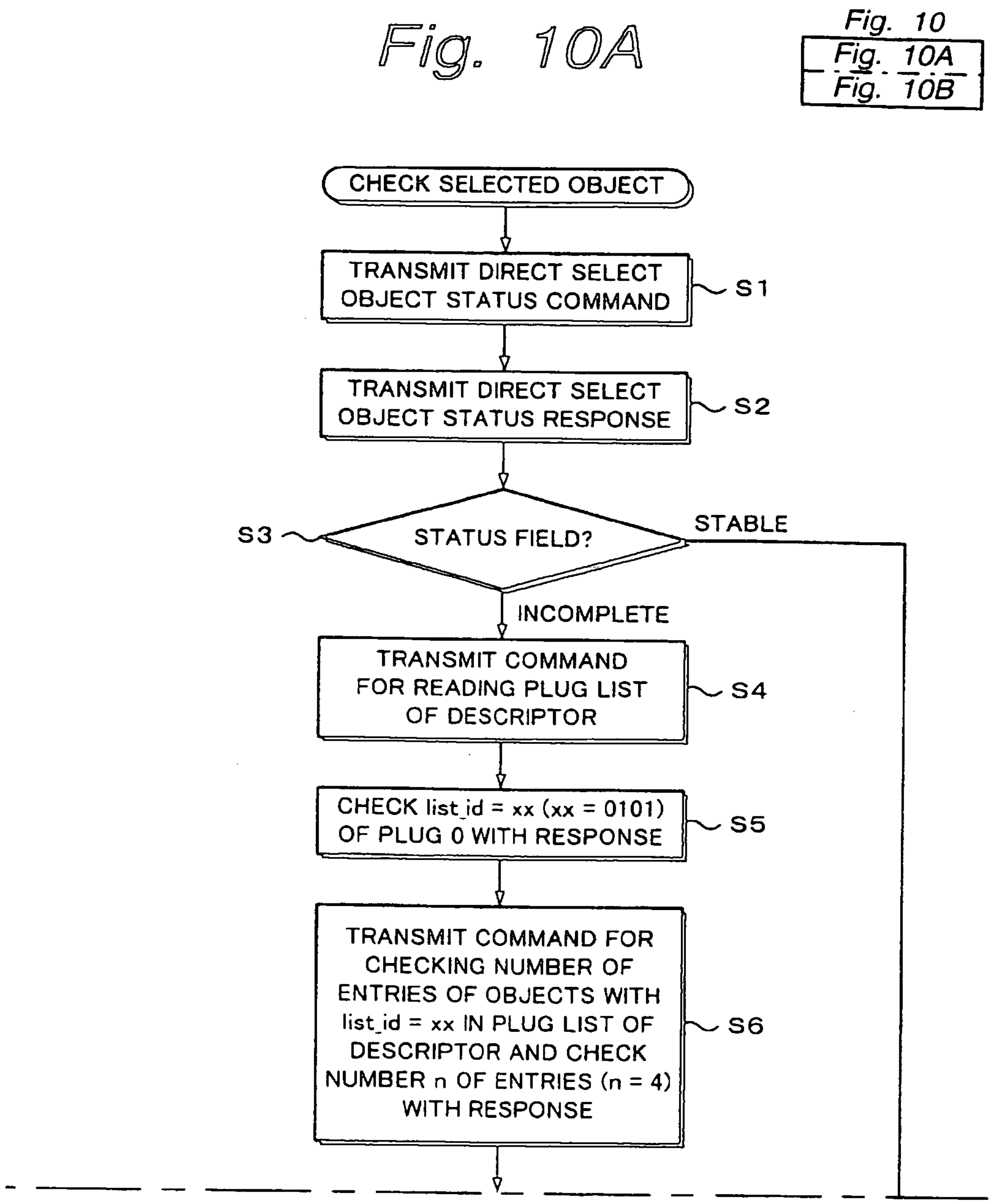


Fig. 10
Fig. 10A
Fig. 10B
CHECK SELECTED OBJECT
TRANSMIT DIRECT SELECT OBJECT STATUS COMMAND
S1
TRANSMIT DIRECT SELECT OBJECT STATUS RESPONSE
S2
S3
STATUS FIELD?
STABLE
INCOMPLETE
TRANSMIT COMMAND FOR READING PLUG LIST OF DESCRIPTOR
S4
CHECK list_id = xx (xx = 0101) OF PLUG 0 WITH RESPONSE
S5
TRANSMIT COMMAND FOR CHECKING NUMBER OF ENTRIES OF OBJECTS WITH list_id = xx IN PLUG LIST OF DESCRIPTOR AND CHECK NUMBER n OF ENTRIES (n = 4) WITH RESPONSE
S6

Fig. 10B

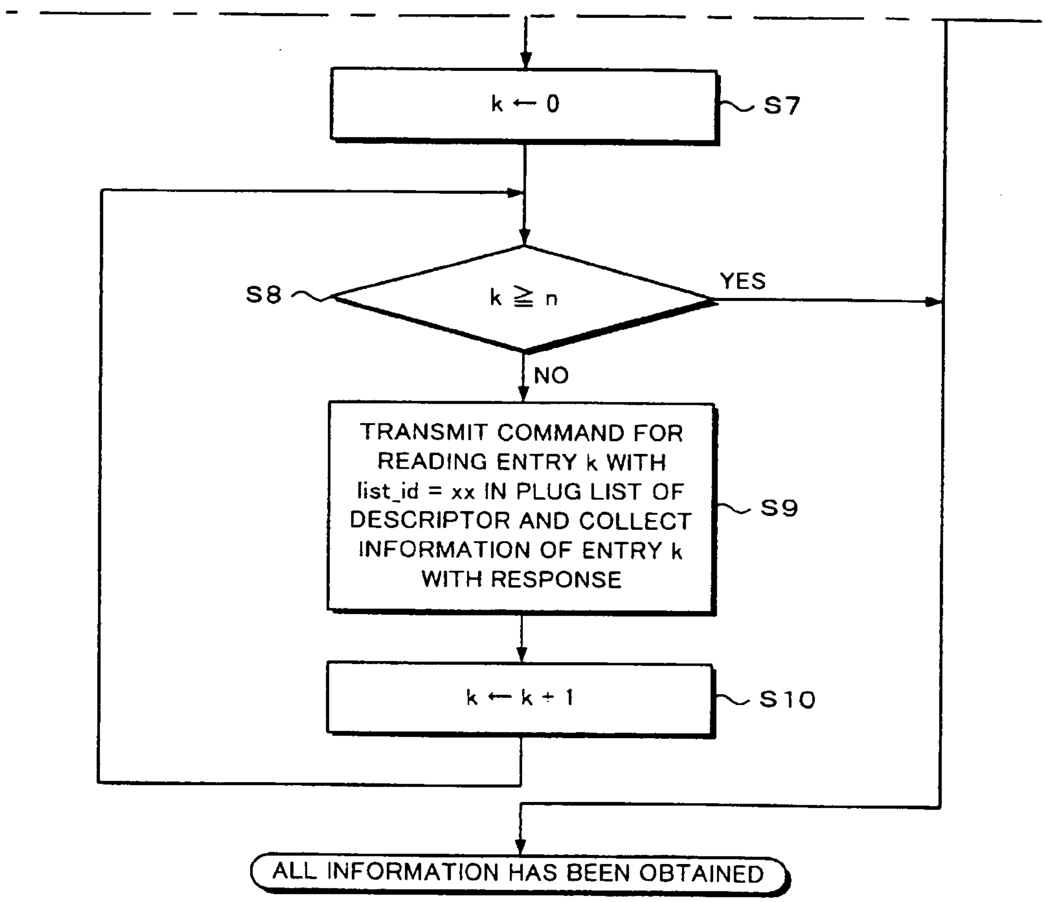

k ← 0
S7
S8
k ≧ n
YES
NO
TRANSMIT COMMAND FOR READING ENTRY k WITH list_id = xx IN PLUG LIST OF DESCRIPTOR AND COLLECT INFORMATION OF ENTRY k WITH RESPONSE
S9
k ← k + 1
S10
ALL INFORMATION HAS BEEN OBTAINED

Fig. 11

opcode	DIRECT SELECT OBJECT
operand [0]	source_plug : plug 0
operand [1]	status : stable
operand [2]	number_of_object_selection_specification : 4
operand [3]	selection_specification [0]
⋮	⋮
operand [x]	selection_specification [3]

Fig. 12

ANTENNA
FREQUENCY
original network id
transport_id
service_id

Fig. 13

opcode	DIRECT SELECT OBJECT
operand [0]	source_plug : plug 0
operand [1]	status : incomplete
operand [2]	number_of_object_selection_specification : 3
operand [3]	selection_specification [0]
	selection_specification [1]
	selection_specification [2]

Fig. 14

opcode		READ DESCRIPTOR	
operand	[0]	data_id	= xx
	[1]		
	[2]	sub_data_id	= kk
	[3]		
	[4]	"FF"	
	[5]	data_length	= 0
	[6]	address	
	[7]		

Fig. 15

opcode		READ DESCRIPTOR	
operand	[0]	data_id	= xx
	[1]		
	[2]	sub_data_id	= k
	[3]		
	[4]	read_result_status	= OK
	[5]	data_length	= yy
	[6]	address	= 0000
	[7]		
	[8]	entry_length	= zz
		data	: ANTENNA
		data	: FREQUENCY
	⋮	⋮	⋮
	[n]	data	